

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/KR04/003510

International filing date: 29 December 2004 (29.12.2004)

Document type: Certified copy of priority document

Document details: Country/Office: KR

Number: 10-2003-0099026

Filing date: 29 December 2003 (29.12.2003)

Date of receipt at the International Bureau: 14 February 2005 (14.02.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

**This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.**

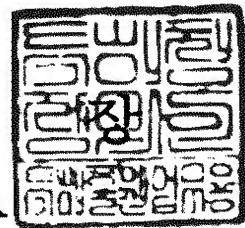
출 원 번 호 : 특허출원 2003년 제 0099026 호
Application Number 10-2003-0099026

출 원 년 월 일 : 2003년 12월 29일
Date of Application DEC 29, 2003

출 원 원 인 : 삼성전자주식회사 외 5명
Applicant(s) SAMSUNG ELECTRONICS CO., LTD., et al.

2005 년 1 월 10 일

특 허 청
COMMISSIONER



【서지사항】

【서류명】	특허 출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2003.12.29
【발명의 명칭】	무선 인터넷 시스템에서의 효율적인 패킷 재전송 방법 및 장치
【발명의 영문명칭】	A method for retransmitting packet in wireless internet system and apparatus thereof
【출원인】	
【명칭】	한국전자통신연구원
【출원인 코드】	3-1998-007763-8
【대리인】	
【명칭】	유미특허법인
【대리인 코드】	9-2001-100003-6
【지정된변리사】	이원일
【포괄위임등록번호】	2001-038431-4
【발명자】	
【성명의 국문표기】	여건민
【성명의 영문표기】	YEO, KUN MIN
【주민등록번호】	691220-1675719
【우편번호】	305-720
【주소】	대전광역시 유성구 신성동 두레아파트 108동 505호
【국적】	KR
【발명자】	
【성명의 국문표기】	윤철식
【성명의 영문표기】	YOON, CHUL SIK
【주민등록번호】	641220-1009115
【우편번호】	139-777
【주소】	서울특별시 노원구 하계동 선경아파트 4동 402호
【국적】	KR
【발명자】	
【성명의 국문표기】	김재홍
【성명의 영문표기】	KIM, JAE HEUNG

【주민등록번호】	660220-1036228		
【우편번호】	305-728		
【주소】	대전광역시 유성구 전민동 세종아파트 106동 807호		
【국적】	KR		
【발명자】			
【성명의 국문표기】	임순용		
【성명의 영문표기】	LIM, SOON YONG		
【주민등록번호】	590315-1017419		
【우편번호】	305-755		
【주소】	대전광역시 유성구 어은동 한빛아파트 117동 1101호		
【국적】	KR		
【발명자】			
【성명의 국문표기】	유병한		
【성명의 영문표기】	RYU, BYUNG HAN		
【주민등록번호】	610205-1807811		
【우편번호】	305-755		
【주소】	대전광역시 유성구 어은동 한빛아파트 118동 604호		
【국적】	KR		
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 유미특 허법인 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	15	면	15,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	0	항	0 원
【합계】	44,000 원		
【감면사유】	정부출연연구기관		
【감면후 수수료】	22,000 원		
【기술이전】			
【기술양도】	희망		
【실시권 허여】	희망		
【기술지도】	희망		
【첨부서류】	1. 요약서·명세서(도면)_1통		

【요약서】

【요약】

본 발명은 이동통신 (또는 무선 인터넷) 시스템에서 패킷 전송의 에러율을 최소화하고 에러 정정 능력을 강화하는 효율적인 ARQ (Automatic Repeat Request) 방식 제공하는 것이다. 이를 위해 ARQ 블록의 상태를 효율적으로 세분화하여 효과적인 스케줄링 및 패킷 전송이 이루어지게 하는 방식, ARQ Feedback 방식 중 Cumulative ACK 방식을 개선한 Bulk ACK/NACK 방법을 고안하여 보다 유연한 ACK/NACK 정보 구성이 가능하도록 방식 및 재전송에 대한 기회부여의 공정성을 확보하기 위해서 타이머 사용 뿐만 아니라 재전송 카운터를 설정하여 2차원적인 공정성 부여가 가능하도록 하는 방법을 제공한다.

【대표도】

도 1

【색인어】

ARQ, 스케줄링, 피드백, 무선 인터넷, 휴대 인터넷

【명세서】

【발명의 명칭】

무선 인터넷 시스템에서의 효율적인 패킷 재전송 방법 및 장치 {A method for retransmitting packet in wireless internet system and apparatus thereof}

【도면의 간단한 설명】

도 1은 ARQ 송신기 및 수신기를 나타낸다.

도 2는 ARQ 송신기 및 수신기에서 사용하는 파라미터들이다.

도 3은 NACK를 받은 경우와 ARQ_RETRY_TIMEOUT을 초과한 경우를 구분하지 않을 때의 ARQ 블록 (block)의 상태를 나타낸다.

도 4는 도 3의 블록 상태 정의를 기반으로 한 ARQ 블록의 상태 천이도 (State transition diagram)을 나타낸다.

도 5는 NACK를 받은 경우와 ARQ_RETRY_TIMEOUT을 초과한 경우를 구분하지 않을 때의 ARQ 블록 (block)의 상태를 나타내며, "Discarded" 상태의 정의가 ARQ_BLOCK_LIFETIME을 초과한 경우와 "Outstanding" 상태에서 NACK를 받았을 때 재전송 회수가 최대 재전송 회수 (ARQ_MAX_RETRANSMIT)를 초과해서 천이된 경우를 포함한다.

도 6은 도 5의 블록 상태 정의를 기반으로 한 ARQ 블록의 상태 천이도 (State transition diagram)을 나타낸다.

도 7은 NACK를 받은 경우와 ARQ_RETRY_TIMEOUT을 초과한 경우를 구분할 때의 ARQ 블록 (block)의 상태를 나타내며, "Retry-timeout" 상태 및 "NACKed" 상태를 추가적으로 가진다.

도 8은 도 7의 블록 상태 정의를 기반으로 한 ARQ 블록의 상태 천이도 (State transition diagram)을 나타낸다.

도 9 NACK를 받은 경우와 ARQ_RETRY_TIMEOUT을 초과한 경우를 구분할 때의 ARQ 블록 (block)의 상태를 나타내며, "Retry-timeout" 상태 및 "NACKed" 상태를 추가적으로 가지며, "Discarded" 상태의 정의가 ARQ_BLOCK_LIFETIME을 초과한 경우와 "Outstanding" 상태에서 NACK를 받았을 때 재전송 회수가 최대 재전송 회수 (ARQ_MAX_RETRASMIT)를 초과해서 천이된 경우를 포함한다.

도 10은 도 9의 블록 상태 정의를 기반으로 한 ARQ 블록의 상태 천이도 (State transition diagram)을 나타낸다.

도 11은 수신측에서 송신측에게 보내는 ARQ Feedback message의 내용을 나타내는 ARQ_Feedback_IE이다

도 12는 상향 링크에서 ACK/NACK 전용 채널 (Dedicated ACK/NACK channel)을 통해서 ACK/NACK를 보낼 때의 ARQ_ACK message format을 나타낸다.

도 13은 상향 링크에서 ACK/NACK 전용 채널 (Dedicated ACK/NACK channel)을 통해서 ACK/NACK를 보낼 때의 ARQ_ACK message format을 나타내며, 여러 ARQ_ACK message들을 연속해서 구성할 수 있다.

도 14는 ACK Map을 나타낸다.

도 15는 도 14에 기반한 Cumulative ACK의 적용 가능 유효 패턴을 예시한다.

도 16은 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타낸다.

도 17은 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타내고, 도 16과 유사하나 각 Bulk의 길이를 부여하는 방식이 다르다.

도 18은 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타내고, 도 17과 유사하나 Bulk Configuration에 따라 Bulk 구성 방법 및 Bulk 개수 그리고 각 Bulk의 길이를 부여하는 방식이 다르다.

도 19는 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타내고, ACK 및 NACK Bulk를 교대로 지정하는 방법 및 ACK/NACK 대상 Bulk의 길이 (ACK Bulk Length, NACK Bulk Length)를 지정하는 방법, 그리고 각 필드에 대해서 비트수를 지정하는 방법을 나타낸다.

도 20은 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타내고, ACK/NACK 대상 Bulk의 길이 및 Next Bulk Flag를 지정하는 방법, 그리고 각 필드에 대해서 비트수를 지정하는 방법을 나타낸다.

도 21은 수신측에서 ARQ 블록의 수신 과정 (ARQ block reception)을 나타낸다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<22> 이동무선환경에서 운용되는 이동통신 (또는 무선 인터넷) 시스템에서 패킷 전송의 에러율을 최소화하고 에러 정정 능력을 강화하는 효율적인 ARQ (Automatic Repeat Request) 방식 제공하기 위한 방안 및 그 장치를 고안한다.

<23> 기존의 ARQ 방식에서는 아래와 같은 문제점 또는 비효율성을 가지고 있다:

<24> 1) 연속적인 블록들로 PDU가 구성되는 상황에서 전송 에러가 발생하면 일련의 연속적인 블록들이 한꺼번에 ACK 또는 NACK인 상황이 자주 발생된다. 이러한 상황에서 Cumulative ACK 방식 (하나의 BSN을 사용하여 해당 BSN의 블록까지 모두 수신에 성공했다고 송신측에 알리는 방식) 및 Cumulative with Selective ACK (Cumulative 방식에 추가하여 이후 블록들에 대해서 ACK/NACK 정보를 비트맵 (bit map) 방식으로 송신측에 알리는 방식)이 효과적으로 적용될 수 있으나, Cumulative ACK 방식은 그 자체로 ARQ Feedback을 주는데 한계가 있으며, Cumulative with Selective 방식은 ARQ Feedback message의 오버헤드를 증가시키는 문제점을 가지고 있다.

<25> 2) 임의 블록이 폐기 조건 (초기 전송 후 ACK를 받기전에 타이머 만료)가 만족되어 폐기 절차가 수행될 경우 Discard message를 수신측에 보내게 되는데, 폐기 상태에 있는 블록이 NACK를 받은 상태에서 폐기 상태로 들어온 것인지 또는 NACK를 받지 못한 상태에서 폐기 상태로 들어온 것인지 구분을 할 수 없다. 구분이 필요한 이유는 NACK를 받은 상태에서 발생한 폐기 절차 이벤트 및 NACK를 받지 못한 상태에서

발생한 폐기 절차 이벤트에 대해서 Discard message를 수신측에 보내는 타이밍을 다르게 할 필요가 있다. 또한 해당 세션의 관련 QoS (Quality of Service) 통계량 중 블록 손실율을 보다 정확하게 계산하는데 있어 구분을 하지 않는 방식과 비교할 때 보다 효율적이다.

<26> 3) 폐기 상태로의 진입 여부를 초기 전송 후 ACK를 받기전에 타이머가 만료되는 조건으로만 설정되어 재전송 회수에 있어 공정성 (fairness)를 확보하지 못한다. 예를 들어, 어떤 블록은 한번 재전송 되어 폐기되고 어떤 블록은 여러 번 재전송 되어 폐기된다면, 블록간 재전송 기회가 동등하다고 볼 수 없다. 최대 재전송 회수의 설정은 효율적인 스케줄링 측면에서 유리하다고 볼 수 있다.

<27> 4) Discard message에 대한 응답이 일반적인 ARQ Feedback message를 통해서 이루어진다면, 폐기 상태에 들어온 블록에 대해서 Discard message 송신여부 및 이에 대한 응답 수신여부에 관계없이 ARQ Feedback message를 통해 ACK만 받으면 바로 폐기할 하는 절차가 버퍼 관리 측면에서 보다 효율적이다.

【발명이 이루고자 하는 기술적 과제】

<28> 본 발명은 전술한 종래 기술에서 제기된 문제점을 해결하기 위하여, 패킷 전송의 에러율을 최소화하고 에러 정정 능력을 강화하는 효율적인 ARQ (Automatic Repeat Request) 방식 제공하는 것이다.

<29> 본 발명이 이루고자 하는 기술적 과제를 구체적으로 열거하면 다음과 같다.

<30> 1) ARQ 블록의 상태를 효율적으로 세분화하여 효과적인 스케줄링 및 패킷 전송이 이루어지게 하였다.

<31> 2) ARQ Feedback 방식 중 Cumulative ACK 방식을 개선하여 이에따른 오버헤드를 줄이고 보다 유연한 ACK/NACK 정보 구성이 가능하도록 하였다.

<32> 3) 재전송에 대한 기회부여의 공정성을 확보하기 위해서 타이머 사용뿐만 아니라 재전송 카운터를 설정하여 2차원적인 공정성 부여가 가능하도록 하였다.

【발명의 구성 및 작용】

<33> 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다.

<34> 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였다. 명세서 전체를 통하여 유사한 부분에 대해서는 동일한 도면 부호를 붙였다. (어떤 부분이 다른 부분과 연결되어 있다고 할 때, 이는 직접적으로 연결되어 있는 경우뿐 아니라 다른 소자를 사이에 두고 전기적으로 연결되어 있는 경우도 포함한다.)

<35> 이제 본 발명의 실시예에 따른 패킷 재전송 방법 및 장치에 대하여 도면을 참고로 하여 상세하게 설명한다.

<36> 도 1은 ARQ 송신기 및 수신기를 나타낸다.

<37> 도 2는 ARQ 송신기 및 수신기에서 사용하는 파라미터들이다.

<38> 도 2에 따른 ARQ 파라미터 중 ARQ_MAX_RETRANSMIT 의 추가 및 그 정의와 세팅 방법이 본 발명의 특징에 따른 실시예중 하나이다.

<39> 도 3은 NACK를 받은 경우와 ARQ_RETRY_TIMEOUT을 초과한 경우를 구분하지 않을 때의 ARQ 블록 (block)의 상태를 나타낸다.

<40> 도 4는 도 3의 블록 상태 정의를 기반으로 한 ARQ 블록의 상태 천이도 (State transition diagram)을 나타낸다. "Outstanding" 상태 또는 "Waiting-for-retransmission" 상태에서 $\text{ark_block_lifetime} > \text{ARQ_BLOCK_LIFETIME}$ 을 초과하면, "Discarded" 상태로 천이한다. "Discarded" 상태에서 ACK를 받으면 "Done" 상태로 천이함과 동시에 해당 블록은 버퍼에서 폐기된다.

<41> 도 4에서 "Discarded" 상태에서 ARQ Discard message 전송 여부에 관계없이 ACK만 받으면 "Done" 상태로 천이하는 과정 및 천이 다이어그램이 본 발명의 특징에 따른 실시예중의 하나이다.

<42> 도 5는 NACK를 받은 경우와 ARQ_RETRY_TIMEOUT을 초과한 경우를 구분하지 않을 때의 ARQ 블록 (block)의 상태를 나타내며, 도 3과 유사하나 "Discarded" 상태의 정의가 $\text{ark_block_lifetime} > \text{ARQ_BLOCK_LIFETIME}$ 을 초과한 경우와 "Outstanding" 상태에서 NACK를 받았을 때 재전송 회수가 최대 재전송 회수 (ARQ_MAX_RETRASMIT)를 초과해서 천이된 경우를 포함한다.

<43> 도 5에서 "Discarded" 상태를 '초기 전송 후 ACK를 받기 전 경과 시간이 ARQ_BLOCK_LIFETIME을 초과한 경우 또는 "Outstanding" 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRASMIT를 초과한 경우'로 정의한 것이 본 발명의 특징에 따른 실시예중의 하나이다.

<44> 도 6은 도 5의 블록 상태 정의를 기반으로 한 ARQ 블록의 상태 천이도 (State transition diagram)을 나타내며, 도 4와 유사하나 "Outstanding" 상태에서 NACK를 받았을 때 재전송 회수가 최대 재전송 회수 (ARQ_MAX_RETRASMIT)를 초과하면 "Discarded" 상태로 천이한다.

<45> 도 6에서 "Discarded" 상태에서 ARQ Discard message 전송 여부에 관계없이 ACK만 받으면 "Done" 상태로 천이하는 과정 및 천이 다이어그램과: 도 6에서 "Outstanding" 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRASMIT를 초과한 경우 "Discarded" 상태로 천이하는 과정 및 천이 다이어그램이 본 발명의 특징에 따른 실시예증의 하나이다.

<46> 도 7은 NACK를 받은 경우와 ARQ_RETRY_TIMEOUT을 초과한 경우를 구분할 때의 ARQ 블록 (block)의 상태를 나타낸다. "Retry-timeout" 상태는 전송 후 ACK를 받기 전 경과 시간이 ARQ_RETRY_TIMEOUT을 초과한 경우를 나타내고 ($ark_retry_timeout > ARQ_RETRY_TIMEOUT$), "NACKed" 상태는 전송 (재전송 포함) 후 NACK를 받은 경우를 나타낸다.

<47> 도 7에서 블록의 상태 정의에서 "Retry-timeout"의 추가 및 그 정의를 '전송 후 ACK를 받기 전 경과 시간이 ARQ_RETRY_TIMEOUT을 초과한 경우'로 한 것과, 도 7에서 블록의 상태 정의에서 "NACKed"의 추가 및 그 정의를 'NACK를 받은 경우'로 한 것이 본 발명의 특징에 따른 실시예증의 하나이다.

<48> 도 8은 도 7의 블록 상태 정의를 기반으로 한 ARQ 블록의 상태 천이도 (Statetransition diagram)을 나타낸다. "Outstanding" 상태, "Retry-timeout" 상태,

또는 "NACKed" 상태에서 $\text{arq_block_lifetime} > \text{ARQ_BLOCK_LIFETIME}$ 을 초과하면, "Discarded" 상태로 전이한다. "Discarded" 상태에서 ACK를 받으면 "Done" 상태로 전이함과 동시에 해당 블록은 버퍼에서 폐기된다.

<49> 도 8에서 "Retry-timeout" 및 "NACKed" 상태 추가와; 도 8에서 "Outstanding" 상태에서 $\text{arq_retry_timeout} > \text{ARQ_RETRY_TIMEOUT}$ 을 초과하면 "Retry-timeout" 상태로 가는 천이과정 및 NACK를 받으면 "NACKed" 상태로 가는 천이과정과; 도 8에서 "Retry-timeout" 상태에서 Retransmit이 발생하면 "Outstanding" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에서 ACK를 받으면 "Done" 상태로 가는 천이과정. 그리고 "Retry-timeout" 상태에서 NACK를 받으면 "NACKed" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에서 $\text{arq_block_lifetime} > \text{ARQ_BLOCK_LIFETIME}$ 을 초과하면 "Discarded" 상태로 가는 천이과정; 도 8에서 "NACKed" 상태에서 Retransmit이 발생하면 "Outstanding" 상태로 가는 천이과정, 그리고 "NACKed" 상태에서 ACK를 받으면 "Done" 상태로 가는 천이과정, 그리고 "NACKed" 상태에서 $\text{arq_block_lifetime} > \text{ARQ_BLOCK_LIFETIME}$ 을 초과하면 "Discarded" 상태로 가는 천이과정; 도 8에서 "Discarded" 상태에서 ARQ Discard message 전송 여부에 관계없이 ACK만 받으면 "Done" 상태로 천이하는 과정; 도 8에서 상기 천이 과정을 포함하는 방법이 본 발명의 특징에 따른 실시예중의 하나이다.

<50> 도 9는 NACK를 받은 경우와 ARQ_RETRY_TIMEOUT을 초과한 경우를 구분할 때의 ARQ 블록 (block)의 상태를 나타내며, 도 7과 유사하나 "Discarded" 상태의 정의가 $\text{arq_block_lifetime} > \text{ARQ_BLOCK_LIFETIME}$ 을 초과한 경우와 "Outstanding" 상태에서

NACK를 받았을 때 재전송 회수가 최대 재전송 회수 (ARQ_MAX_RETRASMIT)를 초과해서 천이된 경우를 포함한다.

<51> 도 9에서 블록의 상태 정의에서 "Retry-timeout"의 추가 및 그 정의를 '전송 후 ACK를 받기 전 경과 시간이 ARQ_RETRY_TIMEOUT을 초과한 경우'로 한 것과, 도 9에서 블록의 상태 정의에서 "NACKed"의 추가 및 그 정의를 'NACK를 받은 경우'로 한 것; 도 9에서 "Discarded" 상태를 '초기 전송 후 ACK를 받기 전 경과 시간이 ARQ_BLOCK_LIFETIME을 초과한 경우 또는 "Outstanding" 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRASMIT를 초과한 경우'로 정의한 것이 본 발명의 특징에 따른 실시예의 하나이다.

<52> 도 10은 도 9의 블록 상태 정의를 기반으로 한 ARQ 블록의 상태 천이도 (State transition diagram)을 나타내며, 도 8과 유사하나 "Outstanding" 상태에서 NACK를 받았을 때 재전송 회수가 최대 재전송 회수 (ARQ_MAX_RETRASMIT)를 초과하면 "Discarded" 상태로 천이한다.

<53> 도 10에서 "Retry-timeout" 및 "NACKed" 상태 추가; 도 8에서 "Outstanding" 상태에서 `ark_retry_timeout` ARQ_RETRY_TIMEOUT을 초과하면 "Retry_timeout" 상태로 가는 천이과정 및 NACK를 받으면 "NACKed" 상태로 가는 천이과정; 도 10에서 "Retry-timeout" 상태에서 Retransmit이 발생하면 "Outstanding" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에서 ACK를 받으면 "Done" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에서 NACK를 받으면 "NACKed" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에서 `ark_block_lifetime` ARQ_BLOCK_LIFETIME을 초과하면 "Discarded"

" 상태로 가는 천이과정 : 도 10에서 "NACKed" 상태에서 Retransmit이 발생하면 " Outstanding" 상태로 가는 천이과정 , 그리고 "NACKed" 상태에서 ACK를 받으면 "Done" 상태로 가는 천이과정 , 그리고 "NACKed" 상태에서 arq_block_lifetime이 ARQ_BLOCK_LIFETIME을 초과하면 "Discarded" 상태로 가는 천이과정 : 도 10에서 " Discarded" 상태에서 ARQ Discard message 전송 여부에 관계없이 ACK만 받으면 "Done" 상태로 천이하는 과정 : 도 10에서 "Outstanding" 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRANSMIT를 초과한 경우 "Discarded" 상태로 천이하는 과정 : 도 10에서 상기 천이 과정을 포함하는 천이 방법이 본 발명의 특징에 따른 실시예증의 하나이다 .

<54> 도 11은 수신측에서 송신측에게 보내는 ARQ Feedback message의 내용을 나타내는 ARQ_Feedback_I[E]다 . ACK Type은 ACK/NACK를 보내는 타입을 정의하며 , 이 필드가 0일때는 "Selective ACK"를 , 1일때는 "Cumulative ACK"를 , 2일때는 Cumulative with Selective ACK"를 나타낸다 . "Number of ACK Maps"은 해당 ACK Type 하에서 ACK Map의 개수를 지정한다 : ACK Type이 1일 때 (Cumulative ACK) , 이 필드가 0이면 ACK Map이 없으며 , 1이면 ACK Map이 한 개 , 2이면 두 개를 나타내고 , BSN은 BSN까지의 블록이 성공적으로 수신되었음을 가리킨다 . ACK Map는 도 12에 정의되어 있다 .

<55> 도 12는 상향 링크에서 ACK/NACK 전용 채널 (Dedicated ACK/NACK channel)을 통해서 ACK/NACK를 보낼때의 ARQ_ACK message format을 나타낸다 . 각 필드의 정의는 도 11과 동일하다 .

<56> 도 13은 상향 링크에서 ACK/NACK 전용 채널 (Dedicated ACK/NACK channel)을 통해서 ACK/NACK를 보낼때의 ARQ_ACK message format을 나타내며 , 도 12와 유사하나

LAST 비트가 추가되어 본 비트 값이 1일 때까지 여러 ARQ_ACK message들을 연속해서
붙일 수 있다.

<57> 도 14는 ACK Map을 나타낸다. ACK Type이 1일 때 (Cumulative ACK), BSN은 도
11, 도 12, 또는 도 13의 BSN이후 성공적으로 수신된 연속적인 블록 리스트 내의 가
장 작은 BSN을 가리킨다. Length는 도 14의 BSN부터 성공적으로 수신된 연속적인 블
록들의 개수를 가리킨다. 도 14는 도 11 ~ 13의 ACK Map으로 사용할 수 있다.

<58> 도 15는 Cumulative ACK의 적용 가능 유효 패턴을 예시한다. 예시된 패턴에 대
해서 Cumulative with Selective ACK를 적용할 경우 4 바이트가 필요한데 비해,
Cumulative ACK with Number of ACK Maps = 1을 적용하면 2 바이트만 필요하게 되어
ARQ Feedback을 전송하는 오버헤드 (overhead)를 상대적으로 줄일 수 있다.

<59> 도 16은 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타낸다. 도
16은 도 11 ~ 13의 ACK Map으로 사용할 수 있다.

<60> 도 16은 ACK Map을 같이 구성하는 방법 및 Cumulative ACK일 경우 3개의 Bulk에
대해서 Type을 지정하는 방법과 ACK/NACK 대상 Bulk의 길이 (First Bulk Length,
Second Bulk Length, Third Bulk Length)를 지정하는 방법, 그리고 각 필드에 대해서
비트수를 지정하는 방법을 도시한 것이다.

<61> 도 17은 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타내고, 도
16과 유사하나 각 Bulk의 길이를 부여하는 방식이 다르다. 도 17은 도 11 ~ 13의 ACK
Map으로 사용할 수 있다.

<62> 도 17은 ACK Map을 구성하는 방법 및 Cumulative ACK일 경우 3개의 Bulk에 대해서 Type을 지정하는 방법과 ACK/NACK 대상 Bulk의 길이 (First Bulk Length, Second Bulk Length, Third Bulk Length)를 지정하는 방법, 그리고 각 필드에 대해서 비트수를 지정하는 방법을 도시한 것이다.

<63> 도 18은 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타내고, 도 17과 유사하나 Bulk Configuration에 따라 Bulk 구성 방법 및 Bulk 개수 그리고 각 Bulk의 길이를 부여하는 방식이 다르다.

<64> 도 18은 ACK Map을 구성하는 방법 및 Cumulative ACK일 경우 Bulk Configuration을 지정하는 방법 및 2개 또는 3개의 Bulk에 대해서 Type을 지정하는 방법과 ACK/NACK 대상 Bulk의 길이 (First Bulk Length, Second Bulk Length, 또는 Third Bulk Length)를 지정하는 방법, 그리고 각 필드에 대해서 비트수를 지정하는 방법을 나타낸 것이다.

<65> 도 18은 도 11 ~ 13의 ACK Map으로 사용할 수 있다.

<66> 도 19는 Bulk Cumulative ACK/NACK를 적용할 수 있는 ACK Map을 나타내고, 상기 ACK Map에서는 ACK 및 NACK Bulk를 교대로 지정하고, 각 ACK/NACK 대상 Bulk의 길이 (ACK Bulk Length, NACK Bulk Length)는 4 비트를 사용한다.

<67> 도 19의 ACK Map을 구성하는 방법 및 Cumulative ACK일 경우 ACK 및 NACK Bulk를 교대로 지정하는 방법 및 ACK/NACK 대상 Bulk의 길이 (ACK Bulk Length, NACK Bulk Length)를 지정하는 방법, 그리고 각 필드에 대해서 비트수를 지정하는 방법은 본 발명의 특징에 따른 실시예이다.

<68> 요약하면, 도 14, 16, 17, 18은 11 ~ 13의 ACK Map으로 사용할 수 있다.

<69> 도 20의 ACK Map을 구성하는 방법 및 Cumulative ACK일 경우 Bulk Configuration을 지정하는 방법, 2개 또는 3개의 Bulk에 대해서 Type을 지정하는 방법과 ACK/NACK 대상 Bulk의 길이 (First Bulk Length, Second Bulk Length, 또는 Third Bulk Length) 및 Next Bulk Flag를 지정하는 방법, 그리고 각 필드에 대해서 비트수를 지정하는 방법은 본 발명의 특징에 따른 실시예이다.

<70> 도 21는 수신측에서 ARQ 블록의 수신 과정 (ARQ block reception)을 나타낸다.

<71> 이상에서 본 발명의 바람직한 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

【발명의 효과】

<72> 본 발명은 이동무선 환경에서 운용되는 이동통신 (또는 무선 인터넷) 시스템에서, 패킷 전송의 에러율을 최소화하고 에러 정정 능력을 강화하는 ARQ (Automatic Repeat Request) 기능 중, 기존의 ARQ Feedback message 전송 방식 및 블록 (block) 상태 관리를 개선하여, ARQ Feedback을 전송하는 오버헤드를 줄이고 각 블록의 전송 기회에 대한 공정성을 강화하고 보다 효율적인 스케줄링 및 전송이 이루어지도록 하는 방안을 제공하도록 한다.

【특허청구범위】

【청구항 1】

패킷 재전송 방법에 있어서 :

ARQ 파라미터 중 ARQ_MAX_RETRANSMIT 의 추가 및 그 정의와 세팅 단계를 포함하는 패킷 재전송 방법 .

【청구항 2】

패킷 재전송 방법에 있어서 ,

"Discarded" 상태에서 ARQ Discard message 전송 여부에 관계없이 ACK만 받으면 "Done" 상태로 천이하는 과정을 포함하는 패킷 재전송 방법

【청구항 3】

패킷 재전송 방법에 있어서 ,

"Discarded" 상태를 '초기 전송 후 ACK를 받기 전 경과 시간이 ARQ_BLOCK_LIFETIME을 초과한 경우 또는 "Outstanding" 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRANSMIT를 초과한 경우'로 정의하는 패킷 재전송 방법 .

【청구항 4】

패킷 재전송 방법에 있어서 ,

"Discarded" 상태에서 ARQ Discard message 전송 여부에 관계없이 ACK만 받으면 "Done" 상태로 천이하는 과정 및 천이 디어그램 . 도 6에서 "Outstanding" 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRANSMIT를 초과한 경우 "Discarded" 상태로 천이하는 과정을 포함하는 패킷 재전송 방법

【청구항 5】

패킷 재전송 방법에 있어서,

블록의 상태 정의에서 "Retry-timeout"의 추가 및 그 정의를 '전송 후 ACK를 받기 전 경과 시간이 ARQ_RETRY_TIMEOUT을 초과한 경우'로 하고, 블록의 상태 정의에서 "NACKed"의 추가 및 그 정의를 'NACK를 받은 경우'로 설정하는 패킷 재전송 방법.

【청구항 6】

패킷 재전송 방법에 있어서,

"Retry-timeout" 및 "NACKed" 상태 추가과정, "Outstanding" 상태에서 `arg_retry_timeout` ARQ_RETRY_TIMEOUT을 초과하면 "Retry_timeout" 상태로 가는 천이과정 및 NACK를 받으면 "NACKed" 상태로 가는 천이과정, "Retry-timeout" 상태에서 `Retransmit` 발생하면 "Outstanding" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에서 ACK를 받으면 "Done" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에서 NACK를 받으면 "NACKed" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에서 `arg_block_lifetime` ARQ_BLOCK_LIFETIME을 초과하면 "Discarded" 상태로 가는 천이과정. 도 8에서 "NACKed" 상태에서 `Retransmit` 발생하면 "Outstanding" 상태로 가는 천이과정, 그리고 "NACKed" 상태에서 ACK를 받으면 "Done" 상태로 가는 천이과정, 그리고 "NACKed" 상태에서 `arg_block_lifetime` ARQ_BLOCK_LIFETIME을 초과하면 "Discarded" 상태로 가는 천이과정. 도 8에서 "Discarded" 상태에서 ARQ Discard message 전송 여부에 관계없이 ACK만 받으면 "Done" 상태로 천이하는 과정을 포함하는 패킷 재전송 방법.

【청구항 7】

패킷 재전송 방법에 있어서,

에서 블록의 상태 정의에서 "Retry-timeout"의 추가 및 그 정의를 '전송 후 ACK를 받기 전 경과 시간이 ARQ_RETRY_TIMEOUT을 초과한 경우'로 한 것. 도 9에서 블록의 상태 정의에서 "NACKed"의 추가 및 그 정의를 'NACK를 받은 경우'로 한 것. 도 9에서 "Discarded" 상태를 '초기 전송 후 ACK를 받기 전 경과 시간이 ARQ_BLOCK_LIFETIME을 초과한 경우 또는 "Outstanding" 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRASNMIT를 초과한 경우'로 정의하는 패킷 재전송 방법.

【청구항 8】

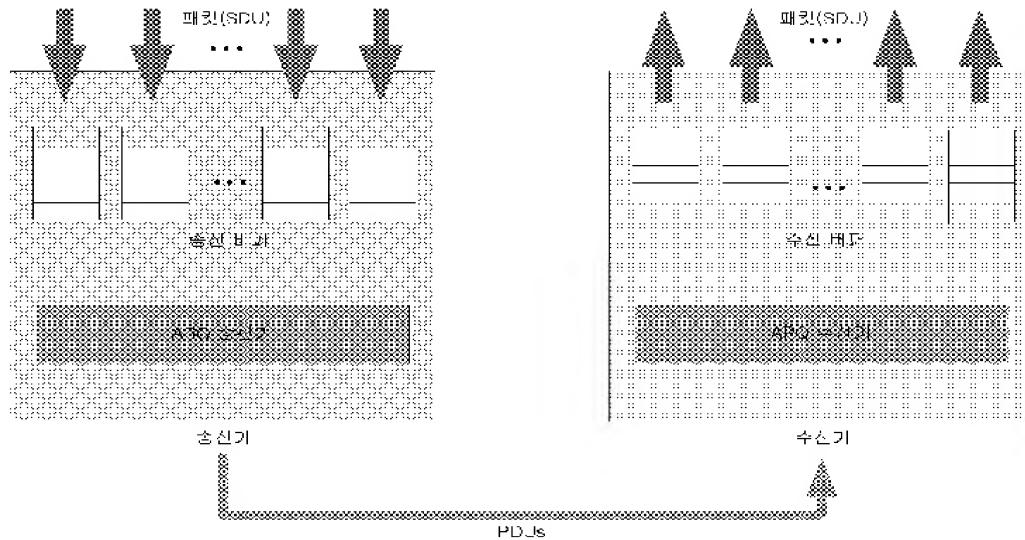
패킷 재전송 방법에 있어서,

"Retry-timeout" 및 "NACKed" 상태 추가과정, "Outstanding" 상태에서
ark_retry_timeout이 ARQ_RETRY_TIMEOUT을 초과하면 "Retry_timeout" 상태로 가는 천
이과정 및 NACK를 받으면 "NACKed" 상태로 가는 천이과정, "Retry-timeout" 상태에서
Retransmit이 발생하면 "Outstanding" 상태로 가는 천이과정, 그리고 "Retry-timeout"
" 상태에서 ACK를 받으면 "Done" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태
에서 NACK를 받으면 "NACKed" 상태로 가는 천이과정, 그리고 "Retry-timeout" 상태에
서 ark_block_lifetime이 ARQ_BLOCK_LIFETIME을 초과하면 "Discarded" 상태로 가는
천이과정. 도 10에서 "NACKed" 상태에서 Retransmit이 발생하면 "Outstanding" 상태로
가는 천이과정, 그리고 "NACKed" 상태에서 ACK를 받으면 "Done" 상태로 가는 천이과정
, 그리고 "NACKed" 상태에서 ark_block_lifetime이 ARQ_BLOCK_LIFETIME을 초과하면 "

"Discarded" 상태로 가는 천이과정 , "Discarded" 상태에서 ARQ Discard message 전송
여부에 관계없이 ACK만 받으면 "Done" 상태로 천이하는 과정 , "Outstanding" 상태에서
NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRASNMIT를 초과한 경우 "Discarded" 상태
로 천이하는 과정을 포함하는 패킷 재전송 방법 .

【도면】

【도 1】



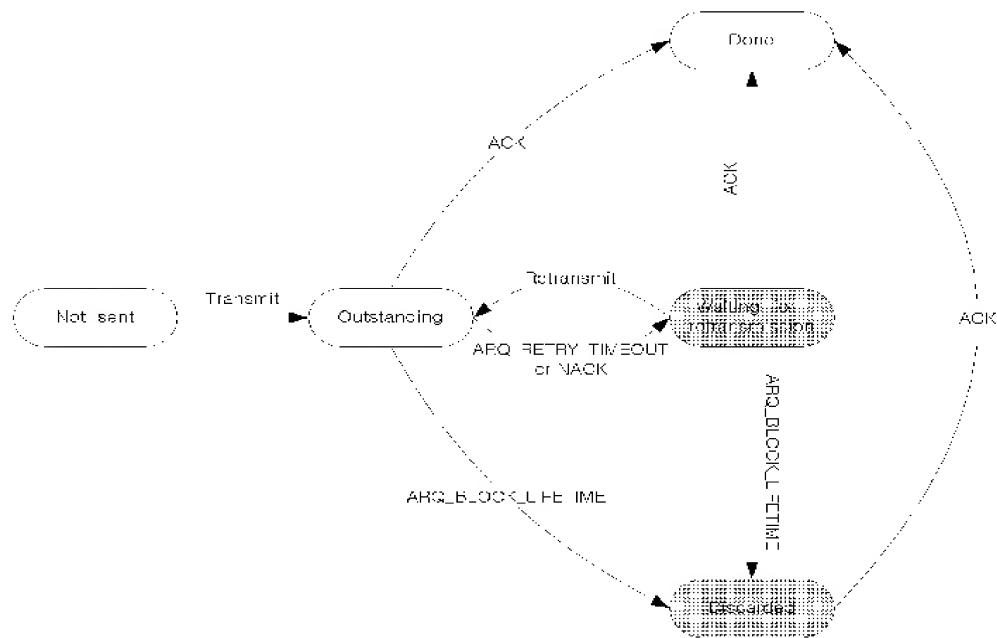
【도 2】

Parameters	Description
<i>ARQ_BSN_MODULUS</i>	The number of unique BSN values: 2^{11}
<i>ARQ_WINDOW_SIZE</i>	The Maximum number of unacknowledged ARQ blocks at any given time: $\leq ARQ_BSN_MODULUS/2$
<i>ARQ_BLOCK_LIFETIME</i>	The maximum time interval an ARQ block shall be managed by the transmitter ARQ state machine, once initial transmission of the block has occurred
<i>ARQ_RETRY_TIMEOUT</i>	The time interval a transmitter shall wait before retransmission of an unacknowledged block for retransmission
<i>ARQ_SYNC LOSS TIME OUT</i>	The maximum time interval <i>ARQ_TX_WINDOW_START</i> or <i>ARQ_RX_WINDOW_START</i> shall be allowed to remain at the same value declaring a loss of synchronization of the sender and receiver state machines when data transfer is known to be active
<i>ARQ_RX_PURGE_TIMEOUT</i>	The time interval the receiver shall wait after successful reception of a block that does not result in advancement of <i>ARQ_RX_WINDOW_START</i> , before advancing <i>ARQ_RX_WINDOW_START</i>
<i>ARQ_BLOCK_SIZE</i>	The size of ARQ block
<i>ARQ_TX_WINDOW_START</i>	All BSNs up to (<i>ARQ_TX_WINDOW_START</i> - 1) have been acknowledged
<i>ARQ_TX_NEXT_BSN</i>	BSN of the next block to send
<i>ARQ_RX_WINDOW_START</i>	All BSNs up to (<i>ARQ_RX_WINDOW_START</i> - 1) have been acknowledged
<i>ARQ_RX_HIGHEST_BSN</i>	BSN of the highest block received + 1
<i>ARQ_MAX_RETRANSMIT</i>	The maximum number of retransmission. This value is set at a connection setup (by DSA (Dynamic Service Addition) message) or connection modification (by DSC (Dynamic Service Change) message). 임의의 블록을 처음으로 전송한 시점부터 현재까지의 시간으로 정의.
<i>Arq_block_lifetime</i>	임의의 블록을 전송한 시점부터 ACK 또는 NACK를 받기 전 현재까지의 시간으로 정의.
<i>Arq_retry_timeout</i>	임의의 블록을 전송한 시점부터 ACK 또는 NACK를 받기 전 현재까지의 시간으로 정의.

【도 3】

Block state	Description
Not-sent	한번도 전송되지 않은 상태
Outstanding	전송중(자전송 포함)
Waiting-for-retransmission	NACK를 받은 경우 또는 전송 후 ACK를 받기 전 경과 시간이 ARQ_RETRY_TIMEOUT을 초과된 경우
Discarded	초기 전송 후 ACK를 받고 저 경과 시간이 ARQ_BLOCK_LIFETIME을 초과된 경우 혹은 “Outstanding” 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRANSMIT를 초과한 경우
Done	ACK를 받은 경우

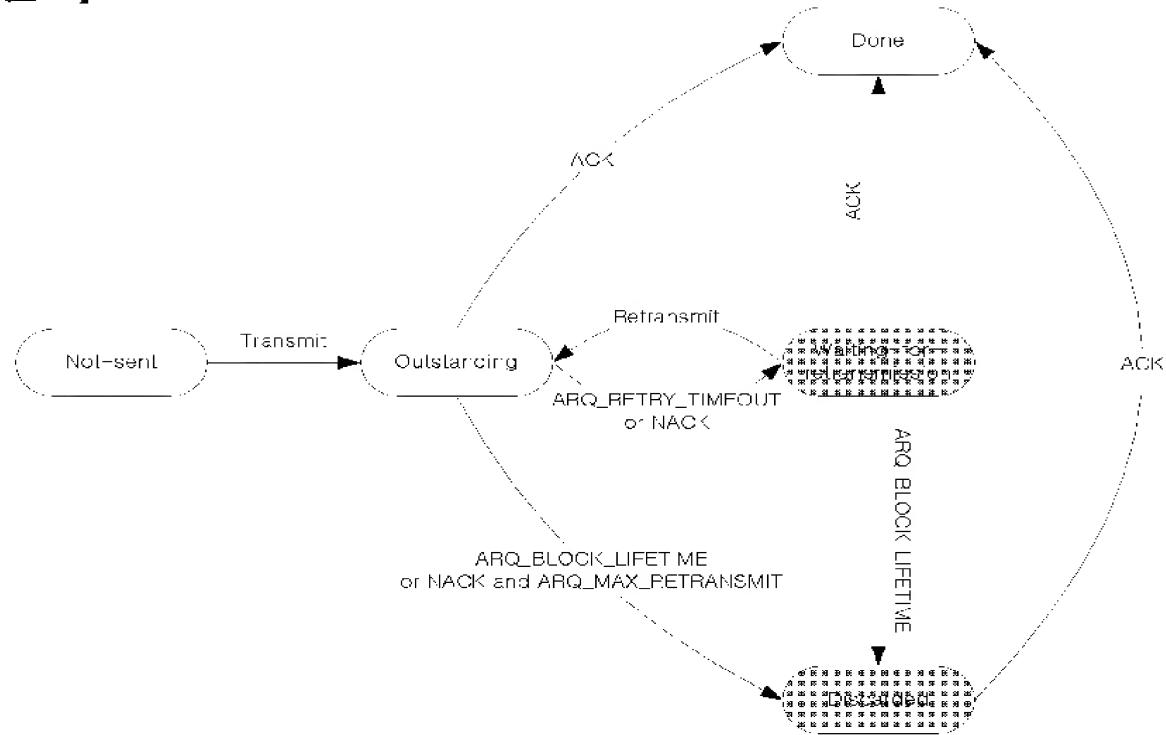
【도 4】



【도 5】

Block state	Description
Not-sent	한번도 전송되지 않은 상태
Outstanding	전송중(자전송 포함)
Waiting-for-retransmission	NACK를 받은 경우 또는 전송 후 ACK를 받기 전 경과 시간이 ARQ_RETRY_TIMEOUT을 초과된 경우
Discarded	초기 전송 후 ACK를 받고 저 경과 시간이 ARQ_BLOCK_LIFETIME을 초과된 경우 혹은 “Outstanding” 상태에서 NACK를 받은 후 재전송 회수가 ARQ_MAX_RETRANSMIT를 초과한 경우
Done	ACK를 받은 경우

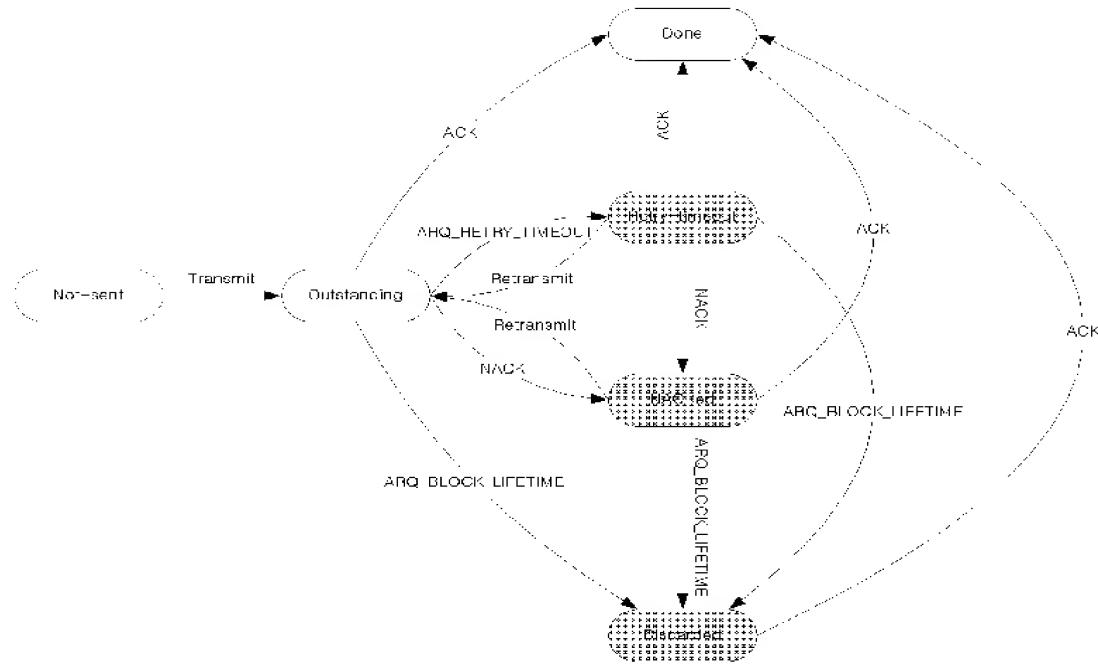
【도 6】



【도 7】

Block state	Description
Not-sent	한번도 전송되지 않은 상태
Outstanding	전송중(재전송 포함)
Retry-timeout	전송 후 ACK를 듣기 전 경과 시간이 ARQ RETRY TIMEOUT을 초과된 경우
NACKed	NACK를 받은 경우
Discarded	초기 전송 후 ACK를 듣기 전 경과 시간이 ARQ BLOCK LIFETIME을 초과된 경우
Done	ACK를 받은 경우

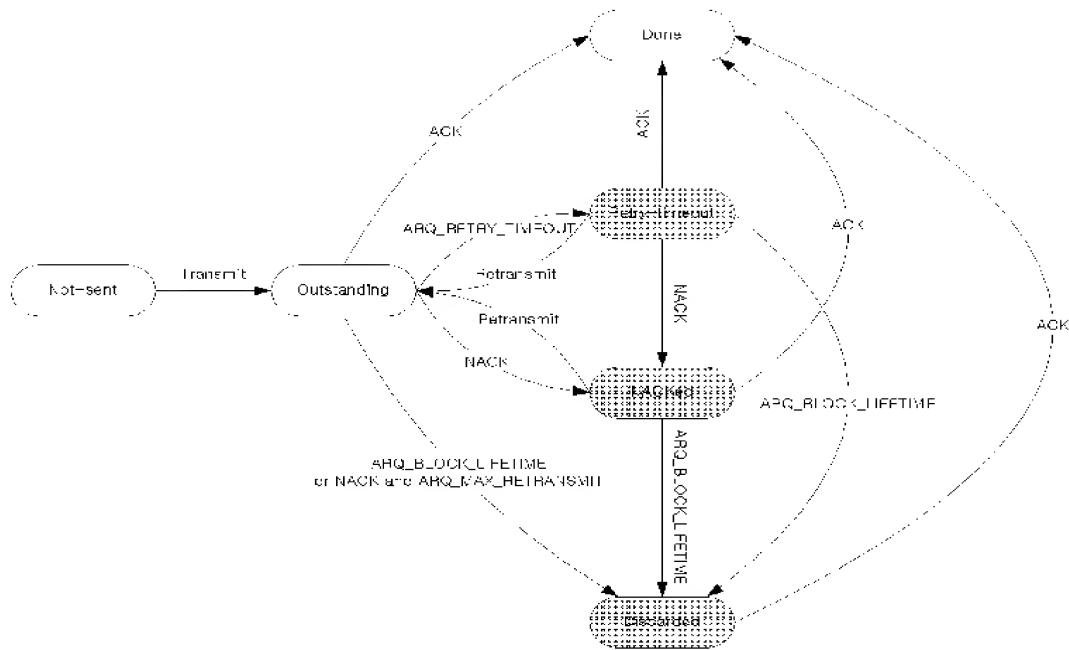
【도 8】



【도 9】

Block state	Description
Not-sent	한번도 전송되지 않은 상태
Outstanding	전송중(자전송 포함)
Retry-timeout	송출 후 ACK를 받기 전 경과 시간이 ARQ_RETRY_TIMEOUT을 초과된 경우
NACKed	NACK를 받은 경우
Discarded	초기 전송 후 ACK를 받기 전 경과 시간이 ARQ_BLOCK_LIFETIME을 초과된 경우 우 모는 “Outstanding” 상태에서 NACK를 받은 후 시선송 외스 ARQ MAY RETRANSMIT를 초과한 경우
Done	ACK를 받은 경우

【도 10】



【도 11】

Syntax	Size	Notes
ARQ_feedback_IE (LAST) {	variable	
CID	16 bits	The ID of the connection being referenced.
LAST	1 bit	0 = More ARQ feedback IE in the list. 1 = Last ARQ feedback IE in the list.
ACK Type	2 bits	0x0 = Selective ACK entry 0x1 = Cumulative ACK 0x2 = Cumulative with Selective ACK entry 0x3 = Reserved
BSN	11 bits	
Number of ACK Maps	2 bits	The field indicates the number of ACK maps: If ACK Type == 01. 0x0 = 0, 0x1 = 1, 0x2 = 2, 0x3= 3; Otherwise. 0x0 = 1, 0x1 = 2, 0x2 = 3, 0x3= 4.
if (ACK Type!= 01) {		
for (i=0; i< Number of ACK Maps + 1; ++i){		
ACK Map	16 bits	This field has different format according to ACK Type. See ACK Map.
}		
}		
}		

【도 12】

Syntax	Size	Notes
ARQ_ACK message format () {	variable	
Reserved	1 bit	
ACK Type	2 bits	0x0 = Selective ACK entry 0x1 = Cumulative ACK 0x2 = Cumulative with Selective ACK entry 0x3 = Reserved
BSN	11 bits	
Number of ACK Maps	2 bits	The field indicates the number of ACK maps: If ACK Type == 01, 0x0 = 0, 0x1 = 1, 0x2 = 2, 0x3 = 3; Otherwise, 0x0 = 1, 0x1 = 2, 0x2 = 3, 0x3 = 4.
if(ACK Type!= 01) {		
for (i=0; i< Number of ACK Maps + 1; ++i)		
}		
ACK Map	16 bits	This field has different format according to ACK Type. See ACK Map.
}		
}		
}		

【도 13】

Syntax	Size	Notes
ARQ_ACK message format(LAST) {	variable	
LAST	1 bit	0 = More ARQ feedback IE in the list. 1 = Last ARQ feedback IE in the list.
ACK Type	2 bits	0x0 = Selective ACK entry 0x1 = Cumulative ACK 0x2 = Cumulative with Selective ACK entry 0x3 = Reserved
BSN	11 bits	
Number of ACK Maps	2 bits	The field indicates the number of ACK maps: If ACK Type == 01, 0x0 = 0, 0x1 = 1, 0x2 = 2, 0x3 = 3; Otherwise, 0x0 = 1, 0x1 = 2, 0x2 = 3, 0x3 = 4.
if(ACK Type!= 01) {		
for (i=0; i< Number of ACK Maps + 1; ++i)		
}		
ACK Map	16 bits	This field has different format according to ACK Type. See ACK Map.
}		
}		
}		

【도 14】

Syntax	Size	Notes
ACK MAP {	16 bits	
if(ACK Type == 01) {		
BSN	11 bits	BSN value indicates that its corresponding block and successive Length (see below) blocks have been successfully received.
Length	5 bits	
}		
else {		
Bit Map	16 bits	In the Bit Map, 1 means that the corresponding block has been successfully received, and 0 means that the corresponding block not been successfully received.
}		
}		

【도 15】

적용 가능한 유형 패턴

... 1111 10[1] 1111 1111 1111 1xxx xxxx ...	1: ACK
... 1111 100[1] 1111 1111 1111 1xxx xxxx ...	0: NACK
... 1111 1000 [1]111 1111 1111 1xxx xxxx ...	x: ACK or NACK
... 1111 1000 0[1]11 1111 1111 1xxx xxxx ...	[1]: First Cumulative ACK end block
... 1111 1000 0000 0000 0000 [1]xxx xxxx ...	[1]: Second Cumulative ACK start block
... 1111 1000 0000 0000 0000 1xxx xxxx ...	

【도 16】

Syntax	Size	Notes
ACK MAP { if (ACK Type == 01) {	16 bits	
Bulk Type	3 bits	Bulk Type indicates the ACK/NACK of the corresponding three bulks (1: ACK, 0: NACK): 1 st bit: ACK/NACK of the first bulk, 2 nd bit: ACK/NACK of the second bulk, 3 rd bit: ACK/NACK of the third bulk.
First Bulk Length	5 bits	The number of blocks (or BSNs) in the first bulk.
Second Bulk Length	4 bits	The number of blocks (or BSNs) in the second bulk.
Third Bulk Length	4 bits	The number of blocks (or BSNs) in the third bulk.
}		
else {		
Bit Map	16 bits	In the Bit Map, 1 means that the corresponding block has been successfully received, and 0 means that the corresponding block not been successfully received.
}		
}		

【도 17】

Syntax	Size	Notes
ACK MAP {	16 bits	
if(ACK Type == 01) {		
Bulk Type	3 bits	Bulk Type indicates the ACK/NACK of the corresponding three bulks (1: ACK, 0: NACK): 1 st bit: ACK/NACK of the first bulk. 2 nd bit: ACK/NACK of the second bulk. 3 rd bit: ACK/NACK of the third bulk.
First Bulk Length	4 bits	The number of blocks (or BSNs) in the first bulk.
Second Bulk Length	4 bits	The number of blocks (or BSNs) in the second bulk.
Third Bulk Length	4 bits	The number of blocks (or BSNs) in the third bulk.
Reserved	1 bit	
}		
else {		
Bit Map	16 bits	In the Bit Map, 1 means that the corresponding block has been successfully received, and 0 means that the corresponding block not been successfully received.
}		
}		

【도 18】

Syntax	Size	Notes
ACK MAP {	16 bits	
if (ACK Type == 01) {		
Bulk Configuration	1 bit	0: the number of bulks is 2 1: the number of bulks is 3
if (Bulk Configuration == 0) {		
Bulk Type	2 bits	Bulk Type indicates the ACK/NACK of the corresponding three bulks (1: ACK, 0: NACK): 1 st bit: ACK/NACK of the first bulk, 2 nd bit: ACK/NACK of the second bulk.
First Bulk Length	6 bits	The number of blocks (or BSNs) in the first bulk.
Second Bulk Length	6 bits	The number of blocks (or BSNs) in the second bulk.
Reserved	1 bits	
}		
Else if (Bulk Configuration == 1) {		
Bulk Type	3 bits	Bulk Type indicates the ACK/NACK of the corresponding three bulks (1: ACK, 0: NACK): 1 st bit: ACK/NACK of the first bulk, 2 nd bit: ACK/NACK of the second bulk, 3 rd bit: ACK/NACK of the third bulk.
First Bulk Length	4 bits	The number of blocks (or BSNs) in the first bulk.
Second Bulk Length	4 bits	The number of blocks (or BSNs) in the second bulk.
Third Bulk Length	4 bits	The number of blocks (or BSNs) in the third bulk.
}		
}		
else {		
Bit Map	16 bits	In the Bit Map, 1 means that the corresponding block has been successfully received, and 0 means that the corresponding block not been successfully received.
}		
}		

【도 19】

Syntax	Size	Notes
ACK MAP {	16 bits	
if (ACK Type == 01) {		
ACK Bulk Length	4 bits	The number of blocks (or BSNs) in the ACK bulk.
NACK Bulk Length	4 bits	The number of blocks (or BSNs) in the NACK bulk.
ACK Bulk Length	4 bits	The number of blocks (or BSNs) in the ACK bulk.
NACK Bulk Length	4 bits	The number of blocks (or BSNs) in the NACK bulk.
}		
else {		
Bit Map	16 bits	In the Bit Map, 1 means that the corresponding block has been successfully received, and 0 means that the corresponding block not been successfully received.
}		
}		

【도 20】

Syntax	Size	Notes
ACK MAP {	16 bits	
if (ACK Type == 01) {		
Bulk Configuration	1 bit	0: the number of bulks is 2 1: the number of bulks is 3
If (Bulk Configuration == 0) {		
First Bulk Length	6 bits	The number of blocks (or BSNs) in the first bulk, the first bulk is always NACK when this ACK MAP is the first entry.
Next Bulk Flag	1 bit	Indicates the ACK/NACK of the next bulk
Second Bulk Length	6 bits	The number of blocks (or BSNs) in the second bulk.
Next Bulk Flag	1 bit	Indicates the ACK/NACK of the next bulk
Reserved	1 bits	
}		
Else if (Bulk Configuration == 1) {		
First Bulk Length	4 bits	The number of blocks (or BSNs) in the first bulk; the first bulk is always NACK when this ACK MAP is the first entry.
Next Bulk Flag	1 bit	Indicates the ACK/NACK of the next bulk
Second Bulk Length	4 bits	The number of blocks (or BSNs) in the second bulk.
Next Bulk Flag	1 bit	Indicates the ACK/NACK of the next bulk
Third Bulk Length	4 bits	The number of blocks (or BSNs) in the third bulk.
Next Bulk Flag	1 bit	Indicates the ACK/NACK of the next bulk
}		
}		
else {		
Bit Map	16 bits	In the Bit Map, 1 means that the corresponding block has been successfully received, and 0 means that the corresponding block not been successfully received.
}		
}		

【도 21】

